

**دانشگاه صنعتی امیرکبیر**  
**( پلی تکنیک تهران )**

دانشکده مهندسی برق

نام دانشجو: کسری خلفی

شماره ی دانشجویی : ۹۵۲۳۰۳۸

استاد درس : دکتر فرزانه عبداللهی

سرپرست آزمایشگاه: مهندس امینی

آزمایش شماره ی ششم

## پایتون:

در ابتدا به پیاده سازی روش هاپفیلد به کمک پایتون میپردازیم. که تابع های مربوط به شکل زیر می باشند:

### ۱- تابع Sign:

در این تابع به کمک numpy اعداد را تشخیص میدهیم که منفی یا مثبت هستند با این تفاوت که به جای صفر عدد -۱ را جایگزین میکنیم

```
6 def sign(a: np.ndarray):
7     w = np.sign(a)
8     w = w.astype(float)
9     for i in range(len(w)):
10         if w[i] == 0:
11             w[i] = -1
12     return w
13
```

### ۲- تابع آسنکرون:

این تابع دارای دو ورودی با نام های v\_new و v\_old می باشد که در خروجی v\_new جدید را به ما برمیگرداند به طوری که تنها یک عبارت از آرایه های وکتور نسبت به قبلی تغییر یابد

```
15 def asyncon(v_new, v_old):
16     a = np.copy(v_old)
17     for i in range(len(v_old)):
18         if a[i] != v_new[i]:
19             a[i] = v_new[i]
20             break
21     return a
22
```

### 3- تابع image\_info\_extractor :

این تابع عکس را میخواند و به صورتی سیاه سفید تبدیل میکند و اعداد بالای ۱۰ به ۲۵۵ و اعداد کوچکتر از ۱۰ به صفر مپ میشوند و در نهایت این تابع را به صورت ستونی تبدیل میکنیم

```

23
24 def image_info_extrackter(name):
25     l1 = cv2.imread(name)
26     l1 = cv2.resize(l1, (10, 10))
27     l1 = cv2.cvtColor(l1, cv2.COLOR_BGR2GRAY)
28     _, l1 = cv2.threshold(l1, 10, 255, 0)
29     l1 = sign(np.reshape(l1, (1, -1))[0])
30     return l1
31
32

```

4- تابع نمایش تصویر:

این تابع با داشتن یک بردار تصویر آن بردار را نمایش میدهد

```

46
47 def show_image_from_info(input: np.ndarray):
48     plt.imshow(input.reshape((10, 10)))
49     plt.show()
50

```

5- تابع نویز:

این تابع به متغیر و وکتور و ارایه ی ما به میزان دلخواه نویز اضافه میکند که ما در این مثال خاص گفتیم که به ارایه ی ۱۰ در ۱۰ ای که داریم که صد تا عدد دارد ۵ تایش را عوض کن و نویزی کند

```

32
33 def noisy(name, percent=10):
34     out = image_info_extrackter(name)
35     l = np.random.randint(100, size=percent).tolist()
36     for i in l:
37         if out[i] == -1:
38             out[i] = 1
39         else:
40             out[i] = -1
41     return out
42

```

6- تابع debug\_print :

این تابع تنها به این دلیل استفاده شد که بتوانیم دو تابع را در کنار هم به صورت همزمان ببینیم

```

42
43 def debug_print(a: np.ndarray, b: np.ndarray):
44     for i in range(len(a)):
45         print(a[i], b[i])
46

```

سپس عکس های داده شده را به عنوان ورودی به تابع می‌دهیم که بدین صورت می‌باشد:

```
51
52 photo_b = image_info_extrackter("photo_b.jpg")
53 photo_d = image_info_extrackter("photo_d.jpg")
54 photo_g = image_info_extrackter("photo_g.jpg")
55 photo_k = image_info_extrackter("photo_k.jpg")
56 photo_p = image_info_extrackter("photo_p.jpg")
57
```

سپس داده ها را نویزی کرده تا با حرف اصلی شباهت نداشته باشند

```
57
58 noisy_b = noisy("photo_b.jpg", 5)
59 noisy_d = noisy("photo_d.jpg", 0)
60 noisy_g = noisy("photo_g.jpg", 5)
61 noisy_k = noisy("photo_k.jpg", 5)
62 noisy_p = noisy("photo_p.jpg", 5)
63
```

سپس  $W$  را به کمک این ماتریس ها به صورت زیر تشکیل می‌دهیم

```
54
55 pattern = np.transpose(np.concatenate([photo_b, photo_d, photo_g, photo_k, photo_p], axis=0))
56 m = pattern.shape[0]
57 n = pattern.shape[1]
58 W = np.dot(pattern, np.transpose(pattern)) - n * np.eye(m)
59
```

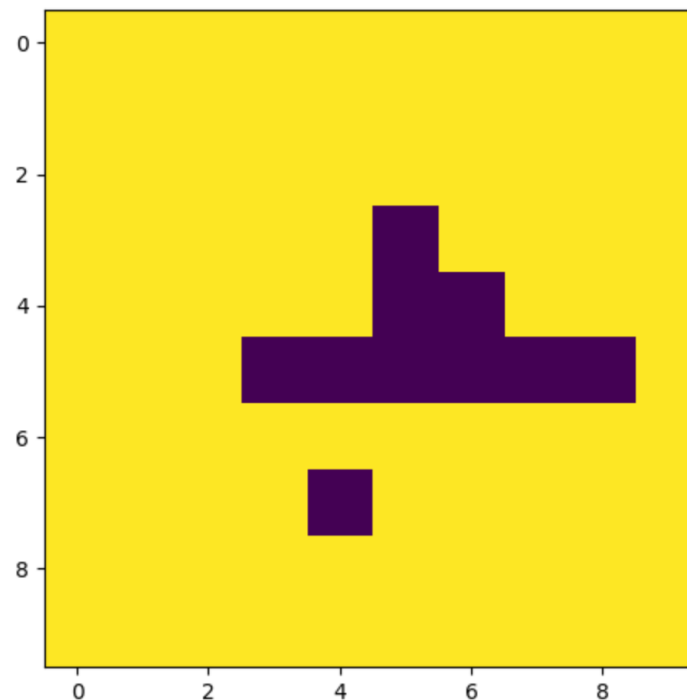
با تشکیل  $W$  و ضرب در ساختار یاد شده را در داخل یک `while` گذاشته و انقدر پیش می‌رویم تا دو داده با هم برابر گردند

```

77
78 input = photo_b
79 show_image_from_info(input)
80 v0 = np.transpose(np.array([input]))
81
82 v_old = v0
83 counter = 0
84 v_new = np.ones((len(v_old), 1))
85
86 while True:
87     v_new_temp = sign(np.dot(W, v_old))
88     debug_print(v_old, v_new_temp)
89     break
90     counter += 1
91     show_image_from_info(v_new)
92     print("iter {} -> {}".format(counter, np.transpose(v_new)))
93     if np.allclose(v_old, v_new):
94         print("result -> {}".format(np.transpose(v_new)))
95         show_image_from_info(v_new)
96         break
97     v_old = v_new
98

```

داده ی ورودی جهت تشخیص به صورت زیر میباشد:



که به ترتیب برای learn ترتیب به صورت زیر میباشد:

